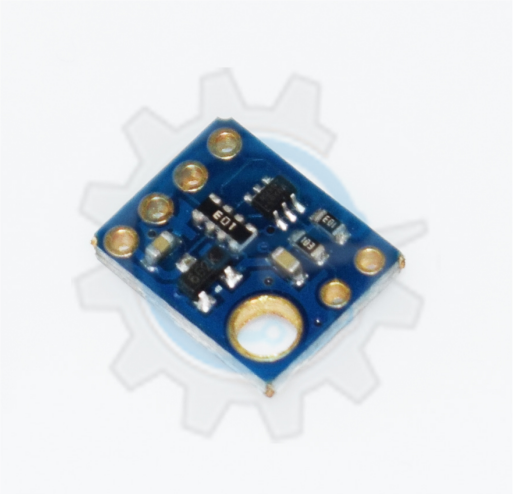
VL53L0X Time-of-Flight Distance sensor



The GY-530 VL53L0X ToF distance sensor is a new generation Time-of-flight (ToF) laser-ranging module housed in the smallest package, providing accurate distance measurement whatever the target reflectances.It can meaure absolute distances up to 2m. Some of this applications is for user detection for personal computers,laptops, IoT (enery saving), robotics (obstacle detection). Compatible in all gizduino boards and Arduino MCU.

Features:

~ Fully integrated miniture module
- 940mm Laser VCSEL
- Ranging sensor with advanced embedded micro controller

~ Fast, accurate distance ranging
- Measures absolute range up to 2m
- Operates in high infrared ambient light levels

~ Eye safe
- Class 1 laser device compilant with latest standard IEC 60825-1:2014. 3rd ed.

~ Easy integration-Single reflowable component
- No additional optics
- Single power supply
- I2C interface for device control and data transfer
- Xshutdown (Reset) and interrupt GPIO
- Programmable I2C Address

Wiring Connections:
GizduinoV to VL53L0X sensor

+5V	**VIN**
GND	**GND**
SCL	**A5/D19**
SDA	**A4/D18**

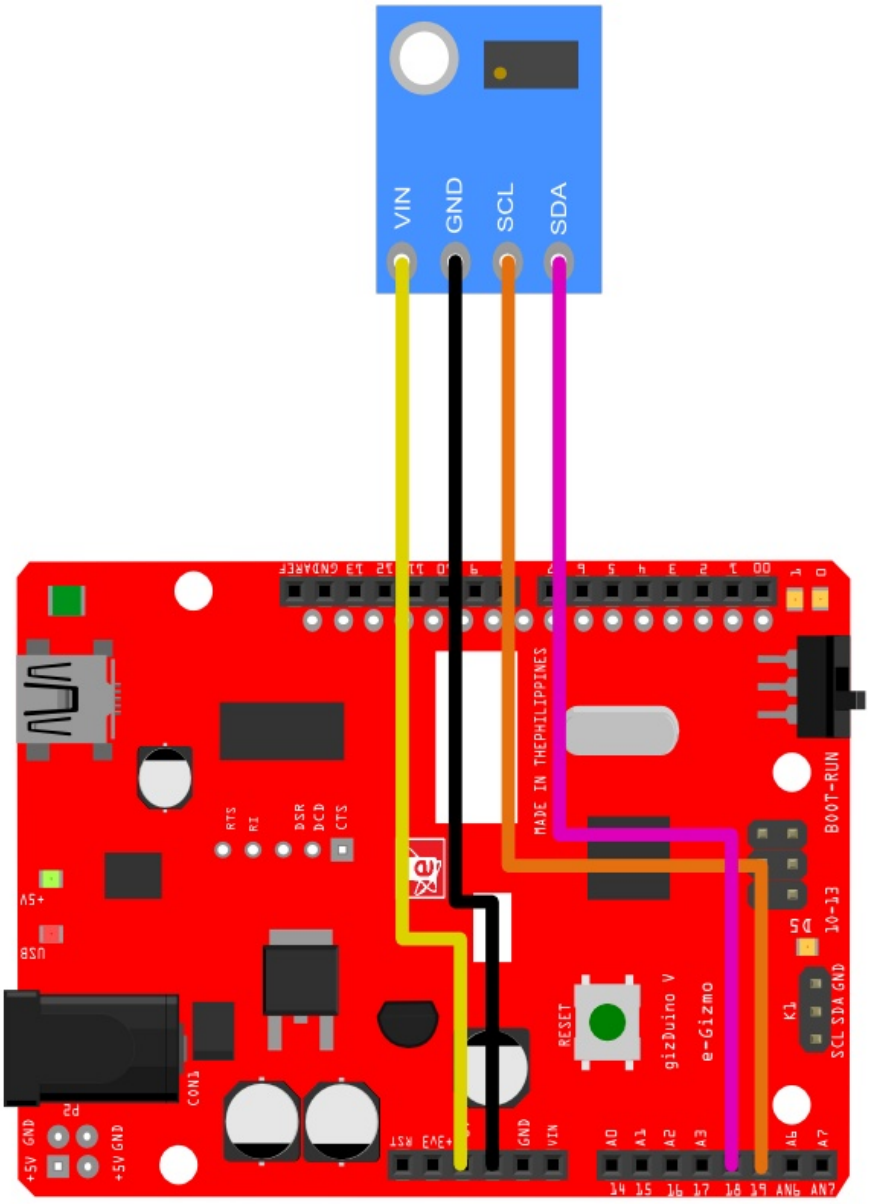


Figure 1. Sample Wiring Diagram with GizDuino V ATmega328P.

```
//*********************************************************//
//                                                         //
//       VL53L0X ToF Distance sensor                       //
//                                                         //
//   This is a sample sketch for GY-530 VL53L-             //
// 0X (Time-Of-Flight) Distance sensor for                 //
// getting the data and display it to the                  //
// Serial Monitor.                                         //
//                                                         //
// the original code by Ted Meyers                         //
// posted here: https://groups.google.com/d                //
// /msg/diyrovers/lc7NUZYuJOg/ICPrYNJGBgAJ                 //
//                                                         //
//  modified by e-Gizmo Mechatronix Central                //
//   11/17/17 www.e-gizmo.net                              //
//*********************************************************//
#include <Wire.h>

#define VL53L0X_REG_IDENTIFICATION_MODEL_ID         0xc0
#define VL53L0X_REG_IDENTIFICATION_REVISION_ID      0xc2
#define VL53L0X_REG_PRE_RANGE_CONFIG_VCSEL_PERIOD   0x50
#define VL53L0X_REG_FINAL_RANGE_CONFIG_VCSEL_PERIOD 0x70
#define VL53L0X_REG_SYSRANGE_START                  0x00
#define VL53L0X_REG_RESULT_INTERRUPT_STATUS         0x13
#define VL53L0X_REG_RESULT_RANGE_STATUS             0x14
#define address 0x29

byte gbuf[16];

void setup() {
  // put your setup code here, to run once:
  Wire.begin();        // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output
  Serial.println("VLX53LOX test started.");
  Serial.println("----- START TEST ----");
  byte val1 = read_byte_data_at(VL53L0X_REG_IDENTIFICATION_REVISION_ID);
  Serial.print("Revision ID: ");
  Serial.println(val1);

  val1 = read_byte_data_at(VL53L0X_REG_IDENTIFICATION_MODEL_ID);
  Serial.print("Device ID: ");
  Serial.println(val1);

  val1 = read_byte_data_at(VL53L0X_REG_PRE_RANGE_CONFIG_VCSEL_PERIOD);
  Serial.print("PRE_RANGE_CONFIG_VCSEL_PERIOD=");
  Serial.println(val1);
  Serial.print(" decode: ");
  Serial.println(VL53L0X_decode_vcsel_period(val1));
```

```
  val1 = read_byte_data_at(VL53L0X_REG_FINAL_RANGE_CONFIG_VCSEL_PERIOD);
  Serial.print("FINAL_RANGE_CONFIG_VCSEL_PERIOD=");
  Serial.println(val1);
  Serial.print(" decode: ");
  Serial.println(VL53L0X_decode_vcsel_period(val1));

}

void loop() {

  test();
  //Serial.println("----- END TEST ----");
  //Serial.println("");
  delay(100);
}

void test() {
  byte val1 = read_byte_data_at(VL53L0X_REG_IDENTIFICATION_REVISION_ID);
  val1 = read_byte_data_at(VL53L0X_REG_IDENTIFICATION_MODEL_ID);
  val1 = read_byte_data_at(VL53L0X_REG_PRE_RANGE_CONFIG_VCSEL_PERIOD);
  val1 = read_byte_data_at(VL53L0X_REG_FINAL_RANGE_CONFIG_VCSEL_PERIOD);

  write_byte_data_at(VL53L0X_REG_SYSRANGE_START, 0x01);

  byte val = 0;
  int cnt = 0;
  while (cnt < 100) { // 1 second waiting time max
    delay(10);
    val = read_byte_data_at(VL53L0X_REG_RESULT_RANGE_STATUS);
    if (val & 0x01) break;
    cnt++;
  }
  if (val & 0x01)// Serial.println("ready");
  // Serial.println("not ready");

  read_block_data_at(0x14, 12);
  uint16_t acnt = makeuint16(gbuf[7], gbuf[6]);
  uint16_t scnt = makeuint16(gbuf[9], gbuf[8]);
  uint16_t dist = makeuint16(gbuf[11], gbuf[10]);
  byte DeviceRangeStatusInternal = ((gbuf[0] & 0x78) >> 3);

  Serial.print("ambient count: ");
  Serial.print(acnt);
  Serial.print("  ");
  Serial.print("signal count: ");
  Serial.print(scnt);
  Serial.print("  ");
  Serial.print("distance: ");
  Serial.print(dist);
```

```
  Serial.print("  ");
  Serial.print("status: ");
  Serial.println(DeviceRangeStatusInternal);
}

uint16_t bswap(byte b[]) {
  // Big Endian unsigned short to little endian unsigned short
  uint16_t val = ((b[0] << 8) & b[1]);
  return val;
}

uint16_t makeuint16(int lsb, int msb) {
  return ((msb & 0xFF) << 8) | (lsb & 0xFF);
}

void write_byte_data(byte data) {
  Wire.beginTransmission(address);
  Wire.write(data);
  Wire.endTransmission();
}

void write_byte_data_at(byte reg, byte data) {
  // write data word at address and register
  Wire.beginTransmission(address);
  Wire.write(reg);
  Wire.write(data);
  Wire.endTransmission();
}

void write_word_data_at(byte reg, uint16_t data) {
  // write data word at address and register
  byte b0 = (data &0xFF);
  byte b1 = ((data >> 8) && 0xFF);

  Wire.beginTransmission(address);
  Wire.write(reg);
  Wire.write(b0);
  Wire.write(b1);
  Wire.endTransmission();
}

byte read_byte_data() {
  Wire.requestFrom(address, 1);
  while (Wire.available() < 1) delay(1);
  byte b = Wire.read();
  return b;
}

byte read_byte_data_at(byte reg) {
  //write_byte_data((byte)0x00);
```

```
  write_byte_data(reg);
  Wire.requestFrom(address, 1);
  while (Wire.available() < 1) delay(1);
  byte b = Wire.read();
  return b;
}

uint16_t read_word_data_at(byte reg) {
  write_byte_data(reg);
  Wire.requestFrom(address, 2);
  while (Wire.available() < 2) delay(1);
  gbuf[0] = Wire.read();
  gbuf[1] = Wire.read();
  return bswap(gbuf);
}

void read_block_data_at(byte reg, int sz) {
  int i = 0;
  write_byte_data(reg);
  Wire.requestFrom(address, sz);
  for (i=0; i<sz; i++) {
    while (Wire.available() < 1) delay(1);
    gbuf[i] = Wire.read();
  }
}


uint16_t VL53L0X_decode_vcsel_period(short vcsel_period_reg) {
  // Converts the encoded VCSEL period register value into the real
  // period in PLL clocks
  uint16_t vcsel_period_pclks = (vcsel_period_reg + 1) << 1;
  return vcsel_period_pclks;
}
```